

# A TOOL FOR SUPPORT OF ONLINE SYMBOLIC COMPUTATIONS

**Katarína Žáková**

*Slovak University of Technology, Faculty of Electrical Engineering and Information Technology  
Ilkovičova 3, 812 19 Bratislava, Slovak Republic  
e-mail: katarina.zakova@stuba.sk*

**Abstract:** The paper presents a possibility how to accomplish numerical and symbolic computations in online applications and in web environment. For this purpose the computer algebra system Maxima was used. The proposed solution is illustrated on the corresponding examples that are oriented to the area of automation and control.

**Keywords:** symbolic calculations, Maxima, web applications

## 1 INTRODUCTION

The expansion of Internet led to the development of web applications in the engineering area, too. Many of them need to accomplish various computations that in general can be numerical and symbolical. In the field of automation these computations are needed for solving of differential equations, computations of control algorithms, identification, signal processing, etc. The symbolic calculations can be used for manipulation with expressions (simplification, factorisation, expansion), for substitution of expressions, symbolic differentiation and integrating, etc.

It would be very reasonable that the developer that develops the web application would be not required to implement all mathematical algorithms but he or she could use an online tool that would be available on the Internet for each interested user. This could save him/her a lot of time. The tool could be useful also for students that would be able to check results of their mathematical, physical or other problems without necessity to instal an additional software on their computer.

There are several possibilities that could be used for the above mentioned purpose. One could start with commercial softwares where maybe the most popular are Maple, Matlab, Mathematica. Each of them has also some kind of the support for Internet use. However, our attention was oriented to the free available software such as Axiom, Maxima, Sage, Yacas, etc. The disadvantage is that they are mostly available only for local installation. Therefore the task was to find a way how to approach them online.

## 2 MAXIMA COMPUTER ALGEBRA SYSTEM

The advantage of the selected Computer Algebra System Maxima consists in the fact that the introduced software is free and offered under GNU GPL license. Its development didn't stop and it can be supposed that it will also continue in the future. Maxima supports TeX format and graphical results can be drawn using Gnuplot, OpenMath, PS or GeomView. The installation is very easy, it is released for MS Windows and for Linux/BSD/Unix operating system as well. The Maxima syntax is simple and it doesn't require a lot time to master it. The user can find similarities with Matlab, Mathematica and Maple that are usually taught and used at

universities. The community around this software is also quite good and a new user can easily incorporate in it.

Except of Maxima the introduced solution is based on cooperation with Gnuplot, LaTeX and dvi2png softwares. Gnuplot is the utility that enables to draw graphical dependences. It supports several types of two-dimensional and three-dimensional graphs. Gnuplot is already included in the Maxima instalation. LaTeX is the typographical system wide-spread mainly in the scientific community. The document files have to be compiled and as the result one gets \*.dvi files that can be converted to the output printing format. The program dvi2png is the convertor that transforms files from dvi to png format. It is used mainly for the conversion of formulas to graphical presentation. In Fig.1 it is possible to see an example of three-dimensional graphics that is generated using Maxima engine. The numerical values for the graph are computed in Maxima, the graph of the function is generated in Gnuplot that is called from Maxima. Then, the conversion to png format is accomplished and the final picture is displayed on the web page.

For the building of the web application we used XHTML, CSS, JavaScript with jQuery library and PHP scripting language.

### 3 SOLUTION DESCRIPTION

To enable to use Maxima as an online application it was necessary to extent its functionality and to build an interface that would enable to approach the program kernel via Internet. The presented tool can be used in two ways:

- in the form of graphical online web environment where the user can solve required numerical or symbolical mathematical tasks,
- in the form of command interface including predefined php functions that can be used in frame of own web application to include the Maxima functionality in own solution.

#### Graphical online web environment

The created graphical user interface is presented in Fig.1. As it was already told it is available online, i.e. from all pc that are connected to Internet.

The user is allowed to select the form of results visualization according to his or her personal preferences. The resulting formulas can be visualized in the form of png pictures or in jsMath format; they can be sorted in ascending or descending order, etc. All these adjustments can be done in Settings part of the program environment.

Fig. 1 shows the Command Window where the user can enter Maxima commands. Results are shown directly below the command area. The commands can be entered one after another one or in a batch file that can be loaded via Import section of the presented environment. Maxima enables to create programs using own native Maxima language or in Lisp programming language. Each language can be used separately or they can also be combined together.

Together with import of files the application allows to export commands and results, too. The user can use 3 possibilities:

- TeX file that contains only outputs in TeX format,
- Maxima file containing only inputs in Maxima format,
- LOG file that contains both inputs and outputs for the later checking of accomplished activity.

From Fig. 1 it is evident that the online environment includes also “Symbolic Vars” and “History” sections. The first one that is also shown in Fig.2 gives information about variables that are stored in the actual workspace (their names can be used in later commands) and the second one records history of realized commands.

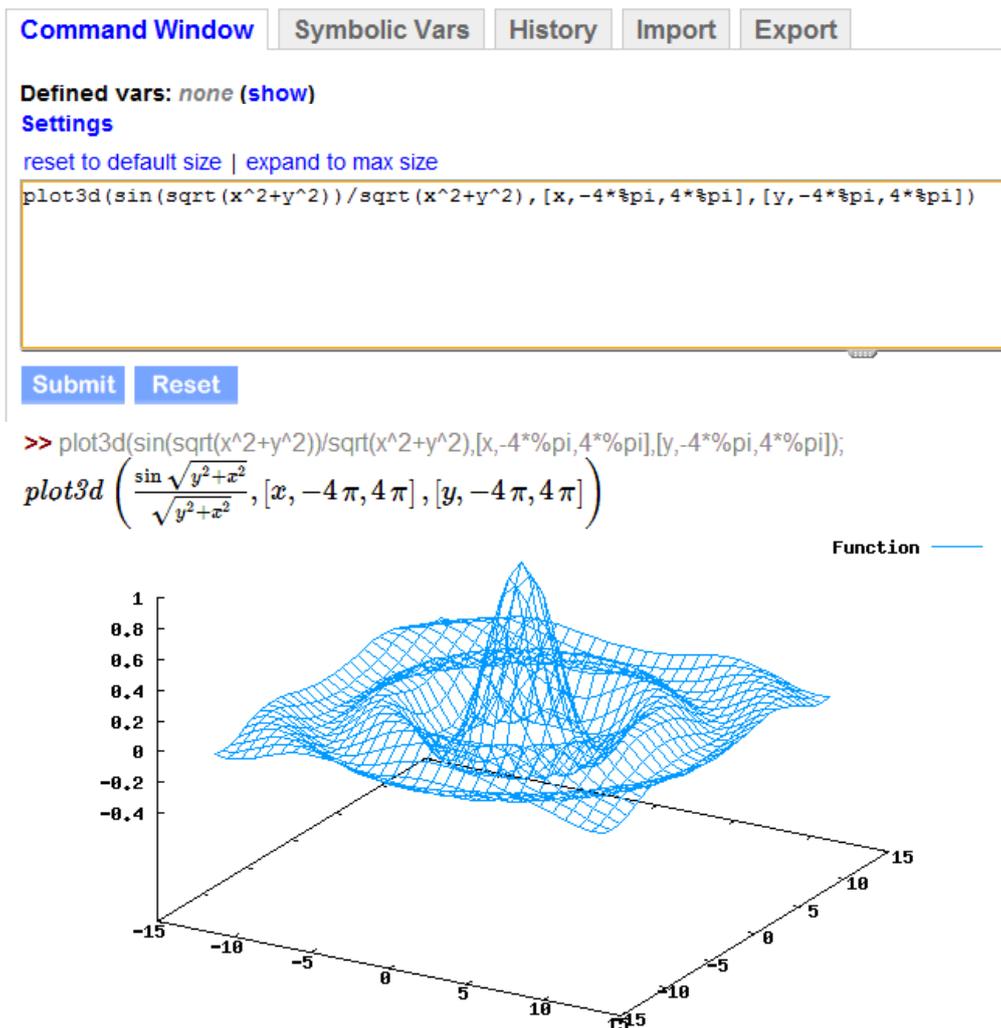


Figure 1: Web interface for Maxima online installation



Figure 2: Visualization of variables in Maxima workspace

The last feature that is to mention is the possibility to generate several versions of the TeX format. It offers a better convenience for user when he or she would like to include the result to the documentation or the scientific paper. All result items enable to open so called “tool tip” from which it is possible to choose the most appropriate pattern for the TeX output. Then, it can be copied to the clipboard by the mouse click or by the well known combination of keys Ctrl+C. The illustration of the mentioned tool tip is shown in Fig.3.

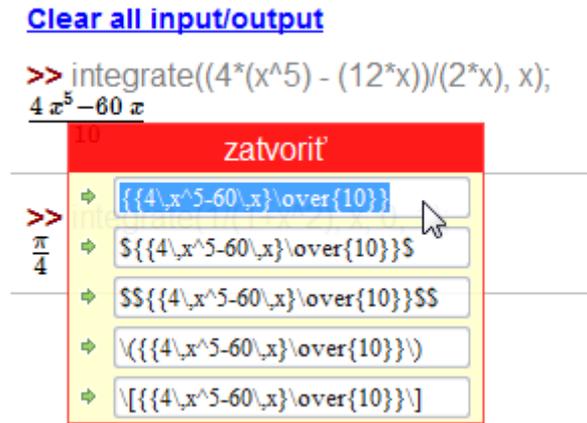


Figure 3: Visualization of variables in Maxima workspace

### Command interface

As it was already introduced the kernel of Maxima can be used not only via the presented online interface. It can be used as an engine in other web applications that require to run more or less complicated mathematical calculations. Actually, the user doesn't need to know about the existence of the Maxima environment on the server at all. In such a case the results from Maxima are sent not to the user but they are sent back to the application that created the operation request. The presented tool includes methods that enable interaction between Maxima and a new web application. These methods allow to send commands to Maxima, to process the Maxima's answer, to diagnostic possible errors and warnings, etc.

The example of the application that uses the Maxima functionality is shown in Fig. 4. The application performs the transformation among several dynamical system descriptions.

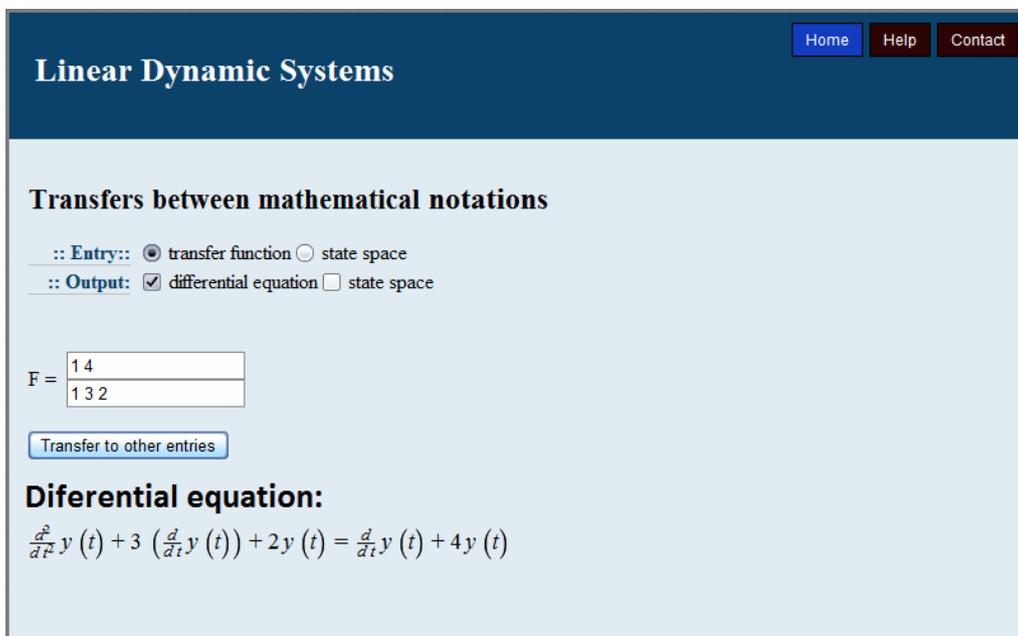


Figure 4: Front-end of the application for model description conversion

## 4 TECHNOLOGY

The created application was tested on the Apache server where all components mentioned in Section 2 were installed. After submitting the request for Maxima (either from the online

environment or from other web application) the commands are sent to the server side PHP script (web-interface.php). This script validates the data and forwards them to another PHP script (php-MAXIMA.php) that is able to send data to Maxima. Before sending it decides whether the data represent the graph plotting command or a single equation command. If the command is the graph plotting command it generates the filename for the image file which is going to carry the image of the plotted graph. After that decision the script creates the shell command string containing Maxima commands and executes it via `shell_exec()` PHP function.

The system shell on the server gets the command and executes it by starting the installed version of Maxima. Then, Maxima runs the accepted batch string. If the batch string contains graph plotting commands it generates the necessary output and call Gnuplot (for 2D and 3D plotting) or Imagemagix (for fractal plotting) applications to generate png image file from Maxima output with the already generated filename. After that Maxima sends its output as the text string back to the shell and the shell forwards it to the PHP script (php-MAXIMA.php).

The PHP script receives the data as the string from the `shell_exec()` function. The script has to remove all unnecessary characters and then the user input and the Maxima output are formatted into the TeX format and sent to other PHP script (web-interface.php) which generates the web page with the Maxima output. Another possibility is that the Maxima output is sent directly to the external web application.

Since the user can specify two formats of the displayed mathematical results (jsMath or png images) the web-interface.php script has to distinguish between both outputs. If the mathematical results should be displayed by jsMath tool the output is generated with the TeX representation of received strings. Otherwise the png images of results are generated using Mathtex and Dvipng tools. Then, all results are sent to the user web browser or to the next application. The described procedure is illustrated in details in Fig. 5.

## 5 CONCLUSIONS

The introduced solution presents an alternative to the classical desktop applications. The main motivation of this work was to design and implement an efficient tool for executing of mathematical calculations that can be used in virtual and remote laboratories. It seems that the tool is able to fulfill this requirement in a very positive way.

## 6 ACKNOWLEDGMENT

The work has been partially supported by the Grant KEGA No. 3/7245/09 and by the Grant VEGA No. 1/0656/09. It was also supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

In addition, author thanks to Dávid Gyurász and Lúdvít Vörös for their ideas, time and all programming work they accomplished on this project.

## REFERENCES

- DODIER, R. (2009): Maxima 5.20.1 Manual.  
<http://maxima.sourceforge.net/docs/manual/en/maxima.html>
- GYURÁSZ, D. (2009): Symbolické výpočty na Internete. Diplomová práca, FEI STU, in Slovak.

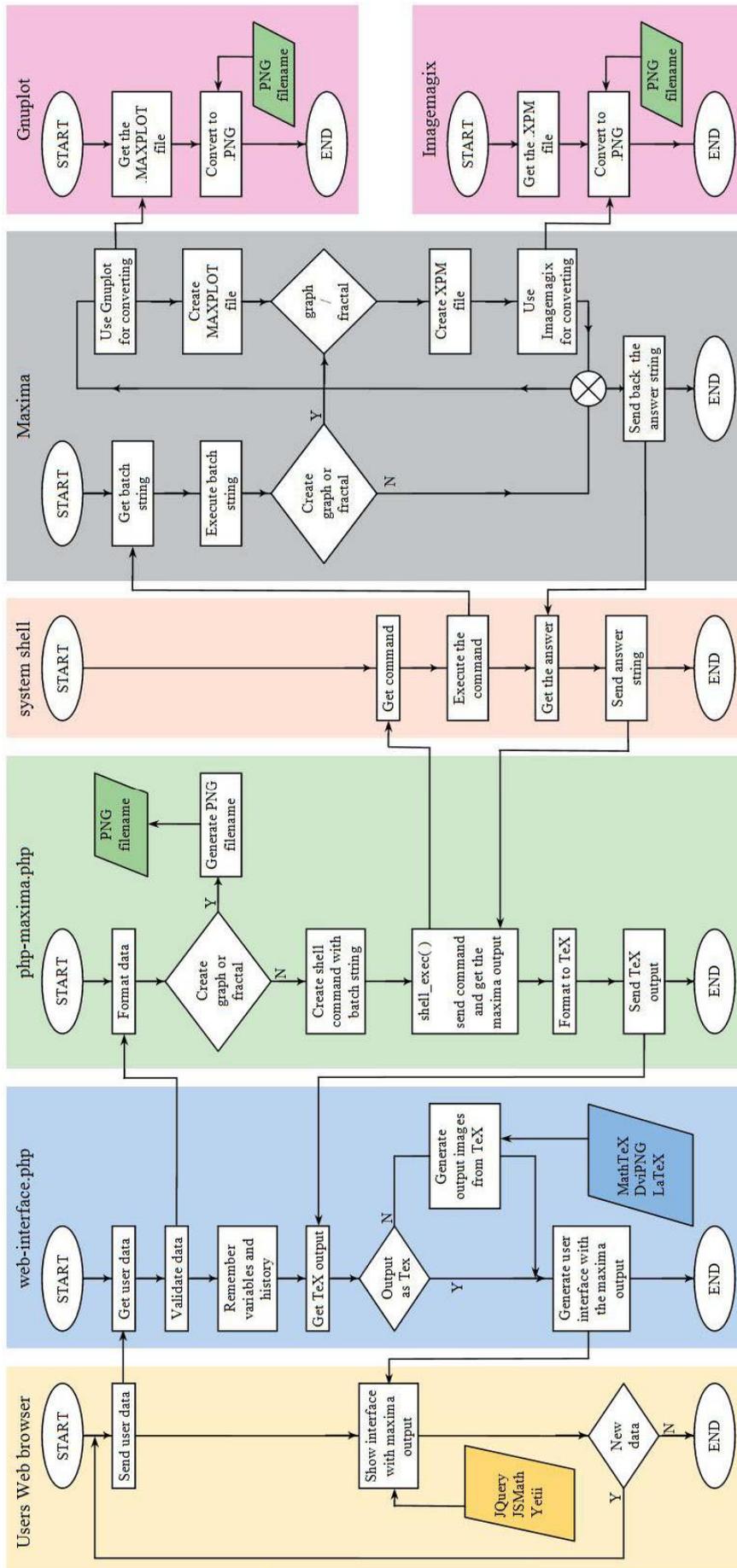


Figure 5: The detailed block scheme of the Maxima implementation on the server

- GAERTNER, B. (2005): Maxima Programming. <http://maxima.sourceforge.net/docs/tutorial/en/gaertner-tutorial-revision/Pages/Programming0001.htm>
- jsMath (2004): A Method of Including Mathematics in Web Pages.  
<http://www.math.union.edu/~dpvc/jsMath/>
- MAGYAR, Z., STARÝ, T., SZOLIK, L., VÖRÖS, L., ŽÁKOVÁ, K. (2009): Modeling of Linear Dynamical Systems using Open Tools. *10th international conference Virtual University*, December, Bratislava.
- MCANDREW, A. (2007): A Morsel of Maxima.  
<http://physics.mercer.edu/young/modern/maxima.pdf>
- SCHMID, CHR. (2003): Internet - basiertes Lernen. *Automatisierungstechnik*, 51, No. 11, pp. 485-493.